

The Connectivity of Boolean Satisfiability: No-Constants and Quantified Variants

Konrad W. Schwerdtfeger

Institut für Theoretische Informatik, Leibniz Universität Hannover,
Appelstr. 4, 30167 Hannover, Germany
`k.w.s@gmx.net`

Abstract. For Boolean satisfiability problems, the structure of the solution space is characterized by the solution graph, where the vertices are the solutions, and two solutions are connected iff they differ in exactly one variable. Motivated by research on heuristics and the satisfiability threshold, Gopalan et al. in 2006 studied connectivity properties of the solution graph and related complexity issues for constraint satisfaction problems in Schaefer’s framework [6]. They found dichotomies for the diameter of connected components and for the complexity of the *st*-connectivity question, and conjectured a trichotomy for the connectivity question that we recently were able to prove.

While Gopalan et al. considered $\text{CNF}(\mathcal{S})$ -formulas with constants, we here look at two important variants: $\text{CNF}(\mathcal{S})$ -formulas without constants, and partially quantified formulas. For the diameter and the *st*-connectivity question, we prove dichotomies analogous to those of Gopalan et al. in these settings. While we cannot give a complete classification for the connectivity problem yet, we identify fragments where it is in P, where it is coNP-complete, and where it is PSPACE-complete, in analogy to Gopalan et al.’s trichotomy.

1 Introduction

In 2006, P. Gopalan, P. G. Kolaitis, E. Maneva, and C. H. Papadimitriou investigated connectivity properties of the solution space of Boolean constraint satisfaction problems (*CSPs*) [7,6]. Their work was motivated inter alia by research on heuristics for satisfiability algorithms and on threshold phenomena [13,12]. Meanwhile, Gopalan et al.’s results have also been applied directly to *reconfiguration problems*, that arise when a step-by-step transformation between two feasible solutions is searched, such that all intermediate results are also feasible [9,10].

The solutions (satisfying assignments) of a formula ϕ over n variables induce a subgraph $G(\phi)$ of the n -dimensional hypercube graph, that is, the vertices are the solutions of ϕ , and two solutions are connected iff they differ in exactly one variable.

Gopalan et al. specifically addressed $\text{CNF}_C(\mathcal{S})$ -formulas ($\text{CNF}(\mathcal{S})$ -formulas with constants), see Definition 2, and studied the complexity of the following two decision problems,

- the *connectivity problem* $\text{CONN}_C(\mathcal{S})$, that asks for a given $\text{CNF}_C(\mathcal{S})$ -formula ϕ whether $G(\phi)$ is connected,
- the *st-connectivity problem* $\text{ST-CONN}_C(\mathcal{S})$, that asks for a given $\text{CNF}_C(\mathcal{S})$ -formula ϕ and two solutions \mathbf{s} and \mathbf{t} whether there a path from \mathbf{s} to \mathbf{t} in $G(\phi)$.

Also, they considered

- the *diameter of ϕ* , that is, the maximal diameter of any connected component of $G(\phi)$ for a $\text{CNF}_C(\mathcal{S})$ -formula ϕ , where the diameter of a component is the maximal shortest-path distance between any two vectors in that component.

They established a common structural and computational dichotomy, and introduced the corresponding class of *tight* sets of relations, which properly contains all Schaefer sets of relations: For tight sets \mathcal{S} , the diameter is linear in the number of variables, $\text{ST-CONN}_C(\mathcal{S})$ is in P and $\text{CONN}_C(\mathcal{S})$ is in coNP, while for all other \mathcal{S} , the diameter can be exponential, and both problems are PSPACE-complete. Moreover, Gopalan et al. conjectured a trichotomy for $\text{CONN}_C(\mathcal{S})$: For a certain sub-class of Schaefer sets of relations, $\text{CONN}_C(\mathcal{S})$ is in P, while for all other tight sets it is coNP-complete; Their results were improved using findings by Makino et al. [11], and recently we could establish the trichotomy [15].

Here, the issues considered by Gopalan et al. are investigated for $\text{CNF}(\mathcal{S})$ -formulas *without* constants and for partially quantified formulas, to the best of our knowledge for the first time. We build on Gopalan et al.’s work [6] (see also the version [8], freely available on ArXiv) and on the paper [15] by the author; the current article will not be comprehensible without reading those papers. Table 1 summarizes our results.

2 Preliminaries

First we introduce some general terminology for Boolean relations and formulas; we will use the standard notions also used in [6] and [15].

Definition 1. An n -ary Boolean relation (or logical relation, relation for short) is a subset of $\{0, 1\}^n$ ($n \geq 1$).

For an n -ary relation R , we can by identification of variables define an $(n - k)$ -ary relation $R'(x_1, \dots, x_{n-k}) = R(\xi_1, \dots, \xi_n)$ ($0 < k < n$), where each $\xi_i \in \{x_1, \dots, x_{n-k}\}$.

The set of solutions of a propositional formula ϕ over n variables defines in a natural way an n -ary relation $[\phi]$, where the variables are taken in lexicographic order.

In the following definition note that we call the formulas with constants $\text{CNF}_C(\mathcal{S})$ -formulas instead of $\text{CNF}(\mathcal{S})$ -formulas as Gopalan et al. Also, for the version of the *st-connectivity* resp. *connectivity* problem with constants, we write $\text{ST-CONN}_C(\mathcal{S})$ resp. $\text{CONN}_C(\mathcal{S})$ instead of $\text{ST-CONN}(\mathcal{S})$ resp. $\text{CONN}(\mathcal{S})$, for consistency with the usual notation $\text{SAT}(\mathcal{S})$ and $\text{SAT}_C(\mathcal{S})$ (see e.g. [14]).

	ST-CONN _C / ST-CONN	CONN _C	CONN	ST-Q-CONN _C	Q-CONN _C
not s.tight, not *	PSPA.-c.	PSPA.-c.	PSPA.-c.	PSPA.-c.	PSPA.-c.
not s.tight, *			??		
s.tight, not *, not Schaefer	in P	coNP.-c.	coNP.-c.		
s.tight, *, not Schaefer			?		
Horn, not m.Horn, not s.c.I-			coNP.-c.		
m.Horn, not s.c.I-		in P	in P	in P	
Horn, s.c.I-, not Q-s.c.I-					
Horn, Q-s.c.I-, not IHSB-					
bijunctive / affine / IHSB-					

Table 1. Summary of our complexity results in comparison to the case considered by Gopalan et al.

* = (0-valid or 1-valid or complementive).

s.tight = safely tight, m.Horn = mixed Horn, s.c.I- = safely componentwise IHSB-.

?: in coNP; may be in P, may be coNP-complete.

??: in PSPACE; may be in P, may be coNP-complete, may be PSPACE-complete.

The cases involving dual Horn, IHSB+ etc. sets of relations are analogous.

The diameter is linear exactly when the related st-connectivity problem is in P, and exponential otherwise.

Definition 2. A CNF-formula is a propositional formula of the form $C_1 \wedge \dots \wedge C_m$ ($1 \leq m < \infty$), where each C_i is a clause, that is, a finite disjunction of literals (variables or negated variables). A k -CNF-formula ($k \geq 1$) is a CNF-formula where each C_i has at most k literals. A Horn (dual Horn) formula is a CNF-formula where each C_i has at most one positive (negative) literal.

For a finite set of relations \mathcal{S} , a $\text{CNF}_C(\mathcal{S})$ -formula over a set of variables V is a finite conjunction $C_1 \wedge \dots \wedge C_m$, where each C_i is a constraint application (constraint for short), i.e., an expression of the form $R(\xi_1, \dots, \xi_k)$, with a k -ary relation $R \in \mathcal{S}$, and each ξ_j is a variable from V or one of the constants 0, 1. A $\text{CNF}(\mathcal{S})$ -formula is a $\text{CNF}_C(\mathcal{S})$ -formula where each ξ_j is a variable in V , not a constant.

We use $\mathbf{a}, \mathbf{b}, \dots$ to denote vectors of Boolean values and $\mathbf{x}, \mathbf{y}, \dots$ to denote vectors of variables, $\mathbf{a} = (a_1, a_2, \dots)$ and $\mathbf{x} = (x_1, x_2, \dots)$.

For the definitions of the relevant classes of relations and sets of relations we refer to Definitions 4 and 9 of [15].

3 No-Constants

In this section we study $\text{CNF}(\mathcal{S})$ -formulas without constants; the respective st-connectivity resp. connectivity problems are denoted by $\text{CONN}(\mathcal{S})$ resp. $\text{ST-CONN}(\mathcal{S})$.

3.1 A Dichotomy for $\text{st-Conn}(\mathcal{S})$ and the Diameter

Gopalan et al. stated in [7] that they could extend their dichotomy theorem for st-connectivity to formulas without constants, without giving the proof. Here we show that for the st-connectivity problem and the diameter, the same dichotomy they proved for formulas with constants indeed also holds for formulas without constants.

Theorem 1. *Let \mathcal{S} be a finite set of logical relations.*

1. *If \mathcal{S} is safely tight, $\text{ST-CONN}(\mathcal{S})$ is in P, and the diameter of $G(\phi)$ for every $\text{CNF}(\mathcal{S})$ -formula ϕ is linear in the number of variables.*
2. *Otherwise, $\text{ST-CONN}(\mathcal{S})$ is PSPACE-complete, and there are $\text{CNF}(\mathcal{S})$ -formulas ϕ such that the diameter of $G(\phi)$ is exponential in the number of variables.*

Proof. The upper bounds obviously carry over from $\text{ST-CONN}_C(\mathcal{S})$, see the correct version¹ (section 3 in [15]) of Theorem 2.9 of [6]. The PSPACE-hardness and the exponential diameter for \mathcal{S} not safely tight follow from Lemma 2 below. \square

The following definition is central to our reductions from the problems for formulas with constants to those for formulas without:

Definition 3. *A solution \mathbf{a} of a formula ϕ is isolated if \mathbf{a} is not connected to any other solution \mathbf{b} in $G(\phi)$. A formula ϕ is*

- 0-isolating (1-isolating) *if it has an isolated solution $\mathbf{a} \neq (1 \cdots 1)$ ($\mathbf{a} \neq (0 \cdots 0)$),*
- 0-unique (1-unique) *if it has a unique solution $\mathbf{a} \neq (1 \cdots 1)$ ($\mathbf{a} \neq (0 \cdots 0)$).*

Similarly we define isolated vectors for relations, and 0-isolating and 1-isolating relations.

Lemma 1. *If an n -ary logical relation R is not safely OR-free, there is a 1-isolating $\text{CNF}(\{R\})$ -formula ϕ .*

Proof. W.l.o.g. assume that OR can be obtained from R by setting the last $n - 2$ coordinates to constants c_3, \dots, c_n (if R is OR-free but not safely OR-free, first obtain a not OR-free relation by identification of variables); then $R(x_1, x_2, c_3, \dots, c_n) = x_1 \vee x_2$.

If $n = 2$, we identify the two coordinates and obtain the 1-isolating relation $\{1\}$.

Else, if all $c_3, \dots, c_n = 1$, we define a 3-ary relation R' by identifying the last $n - 2$ coordinates. Then $R'(x_1, x_2, 1) = x_1 \vee x_2$, and identifying the first two coordinates of R' yields a 2-ary relation R'' with $11 \in R''$ and $01 \notin R''$, thus R'' equals $\{11, 00, 10\}$, $\{11, 00\}$, $\{11, 10\}$ or $\{11\}$. The second and fourth relation are already 1-isolating, the first is $x \vee \overline{y}$, and we obtain a 1-isolating relation

¹ There are small mistakes in Gopalan et al.'s paper that lead to a shift of the boundaries; they are corrected in [15].

by taking $[R(x, y) \wedge R(y, x)] = \{11, 00\}$, and from the third we obtain $\{1\}$ by identifying the two coordinates.

Similarly, if all $c_3, \dots, c_n = 0$, by identifying the last $n - 2$ coordinates, and then the first two, we get a relation R'' with $10 \in R''$ and $00 \notin R''$, thus R'' equals $\{10, 01, 11\}$, $\{10, 01\}$, $\{10, 11\}$ or $\{10\}$. Here again, the second and fourth relation are already 1-isolating, and from the first as well as from the third we obtain $\{1\}$ by identifying the two coordinates.

Otherwise, we define a 3-ary relation R'' by identifying all coordinates i of R with $c_i = 0$, then all with $c_i = 1$, and then the first two. Formally $R''(x_1, x_2, x_3) = R(x_1, x_1, \xi_3, \dots, \xi_n)$, where $\xi_i = x_2$ if $c_i = 1$ and $\xi_i = x_3$ if $c_i = 0$. Then $110 \in R''$ and $010 \notin R''$, and R'' is one of 64 possible relations; Figure 1 shows how to produce a 1-isolating relation from each of them by identification of variables and conjunction. \square

Lemma 2. *If a finite set of logical relations \mathcal{S} is not safely tight, $\text{ST-CONN}_C(\mathcal{S}) \leq_m^p \text{ST-CONN}(\mathcal{S})$, and for every $\text{CNF}_C(\mathcal{S})$ -formula ϕ with n variables and diameter d , there is a $\text{CNF}(\mathcal{S})$ -formula ϕ' with $O(n)$ variables and diameter $d' \geq d$.*

Proof. Since \mathcal{S} is not safely tight, there is some relation $R \in \mathcal{S}$ that is not safely OR-free, and we can obtain a not OR-free relation by identification of variables, and then construct an n -ary 1-isolating relation R_1 by Lemma 1. Similarly, we can construct a m -ary 0-isolating relation R_0 from a not safely NAND-free relation. Let $\mathbf{a} \neq (1 \dots 1)$ be an isolated vector of R_0 , and $\mathbf{b} \neq (0 \dots 0)$ an isolated vector of R_1 ; w.l.o.g. assume $a_1 = 0$ and $b_1 = 1$ (else obtain appropriate relations by permutation of variables).

Now let $\phi(x_1, \dots, x_n)$ be any $\text{CNF}_C(\mathcal{S})$ -formula and \mathbf{s} and \mathbf{t} two solutions of ϕ . We construct a $\text{CNF}(\mathcal{S})$ -formula ϕ' by replacing every occurrence of the constant 0 in ϕ with a new variable y_1 , and every occurrence of the constant 1 with a new variable z_1 , and appending $\wedge R_0(y_1, y_2, \dots, y_m) \wedge R_1(z_1, z_2, \dots, z_n)$ to ϕ (where y_2, \dots, y_m and z_2, \dots, z_n are further new variables). Then $\mathbf{s} \cdot \mathbf{a} \cdot \mathbf{b}$ and $\mathbf{t} \cdot \mathbf{a} \cdot \mathbf{b}$ are connected in $G(\phi')$ iff \mathbf{s} and \mathbf{t} are connected in $G(\phi)$, and the statement for the diameter is now also obvious. \square

3.2 Towards a Trichotomy for $\text{Conn}(\mathcal{S})$: Extension of the Tractable Class

For the connectivity problem, disallowing constants makes a difference. $\text{CONN}_C(\mathcal{S})$ is in P exactly if \mathcal{S} is CPSS, assuming $P \neq \text{coNP}$ (Theorem 12 of [15]), i.e., if \mathcal{S} is bijunctive, Horn and safely componentwise IHSB $-$, dual Horn and safely componentwise IHSB $+$, or affine. For $\text{CONN}(\mathcal{S})$, there are two additional classes of relations for which we can give a polynomial-time algorithm, intersecting the Horn and safely componentwise IHSB $-$, resp. dual Horn and safely componentwise IHSB $+$ classes:

Definition 4. *A logical relation R is mixed Horn (mixed dual Horn) if it is the set of solutions of a Horn formula in which all clauses of size greater than 1 have at least one positive literal (at least one negative literal).*

Example 1. There are mixed Horn relations that are not safely componentwise IHSB-, not bijective and not affine, e.g. $x \vee \bar{y} \vee \bar{z}$.

Lemma 3. *If a finite set of logical relations \mathcal{S} is mixed Horn or mixed dual Horn, there is a polynomial-time algorithm for $\text{CONN}(\mathcal{S})$.*

Proof. We show the proof for \mathcal{S} being mixed Horn, the mixed dual Horn case is analogous. The following algorithm decides for any $\text{CNF}(\mathcal{S})$ -formula ϕ whether $G(\phi)$ is connected. As in Lemma 4.13 of [6], first assign all variables that can take only one value in any solution of ϕ that value. This is possible in polynomial-time since satisfiability for Horn formulas is in P, and produces a connectivity-equivalent formula ϕ' without unit clauses. Now by Lemma 17 of [15], $G(\phi)$ is connected iff ϕ' has a non-empty maximal self-implicating set since ϕ' contains no restraints (see Definition 14 of [15]).

The following polynomial-time algorithm finds a maximal self-implicating set of ϕ' if one exists.

- Let U be the set of all variables that occur as the positive literal in any clause of ϕ' . Repeat the following as long as variables are removed: Remove a variable from U if it only occurs as the positive literal in clauses with not all negated variables from U .

Now U is obviously a maximal self-implicating set of ϕ' , and it is easy to see that the algorithm produces a non-empty maximal self-implicating set if one exists. \square

3.3 Towards a Trichotomy for $\text{Conn}(\mathcal{S})$: coNP-Completeness

In this subsection we prove that $\text{CONN}(\mathcal{S})$ is coNP-complete for all remaining Schaefer sets of relations:

Lemma 4. *If \mathcal{S} is a finite set of Horn relations and contains at least one relation that is not mixed Horn, and at least one relation that is not safely componentwise IHSB-, $\text{CONN}(\mathcal{S})$ is coNP-complete.*

Proof. We show that $\text{CONN}(\{R\})$ is coNP-hard for $R = M$, $R = M \times \{0\}$, $R = M \times \{1\}$ and $R = M \times \{01\}$ with $M = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$ by modifying the proof of Lemma 18 of [15]. Then the statement follows from Lemma 7.

In the proof of Lemma 18 of [15] we can express $\bar{x} \vee y$ (resp. $(\bar{x} \vee y) \times \{0\}$, $(\bar{x} \vee y) \times \{1\}$, or $(\bar{x} \vee y) \times \{01\}$) as $R(x, x, y)$. Also, by Lemma 6, we can express $\bar{x} \vee \bar{y}$ (resp. $(\bar{x} \vee \bar{y}) \times \{0\}$, $(\bar{x} \vee \bar{y}) \times \{1\}$, or $(\bar{x} \vee \bar{y}) \times \{01\}$) from R . In the construction of ϕ we can then use $(\bar{x} \vee y) \times \{0\}$, $(\bar{x} \vee y) \times \{1\}$, or $(\bar{x} \vee y) \times \{01\}$ instead of $\bar{x} \vee y$, as well as $(\bar{x} \vee \bar{y}) \times \{0\}$, $(\bar{x} \vee \bar{y}) \times \{1\}$, or $(\bar{x} \vee \bar{y}) \times \{01\}$ instead of $\bar{x} \vee \bar{y}$, and $M \times \{0\}$, $M \times \{1\}$, or $M \times \{01\}$ instead of M since the connectivity of ϕ is not affected by these replacements. \square

Lemma 5. *If a logical relation R is Horn but not mixed Horn, at least one of the relations $\{0\}$ or $\{01\}$ can be obtained from R by identification and permutation of variables.*

Proof. Since R is not mixed Horn, every CNF-formula representation of R contains at least one constraint of the form $\overline{x_1} \vee \dots \vee \overline{x_k}$, thus $(1 \dots 1) \notin R$. Also, R is not empty (else it were $x \wedge \overline{x}$), so there must be some vector $\mathbf{a} \in R$ with some $a_i = 0$. If $(0 \dots 0) \in R$, we identify all coordinates and obtain $\{0\}$.

Otherwise, we define the relation R' by identifying all coordinates i with $a_i = 0$, and then all with $a_i = 1$. This gives $\{01\}$ or $\{10\}$, and with an appropriate ordering of the coordinates $\{01\}$: Since $(0 \dots 0)$ and $(1 \dots 1)$ are not in R , $\{00\}$ and $\{11\}$ are not in R' . Further, if $\{01\} \in R'$, $\{10\} \notin R'$: Since R is Horn, it is closed under $x \wedge y$ (see e.g. [5, Lemma 4.8]), and so the “to \mathbf{a} complementary” vector $\mathbf{b} = \mathbf{a} \oplus \mathbf{1}$ is not in R , else $\mathbf{a} \wedge \mathbf{b} = (0 \dots 0)$ were in R (where \oplus and \wedge are applied coordinate-wise). \square

Lemma 6. *If a logical relation R is Horn but not mixed Horn, at least one of the relations NAND , $\text{NAND} \times \{0\}$, $\text{NAND} \times \{1\}$ or $\text{NAND} \times \{01\}$ with $\text{NAND} = \overline{x} \vee \overline{y}$ can be represented as a $\text{CNF}(\{R\})$ -formula.*

Proof. Let ϕ be any CNF-formula representation of R . It is easy to see that we can simplify ϕ s.t. the following conditions hold:

- (a) *no redundant restraints:* remove a restraint $\overline{y_1} \vee \dots \vee \overline{y_k}$ if there is another restraint $\overline{y_{i_1}} \vee \dots \vee \overline{y_{i_l}}$ with $\{y_{i_1}, \dots, y_{i_l}\} \subset \{y_1, \dots, y_k\}$.
- (b) *no implications among the variables of a restraint:* remove a variable x from a restraint $\overline{y_1} \vee \dots \vee \overline{y_k}$ if there is an implication $x \vee \overline{y_{i_1}} \vee \dots \vee \overline{y_{i_l}}$ with $\{x, y_{i_1}, \dots, y_{i_l}\} \subset \{y_1, \dots, y_k\}$.

The following steps generate NAND , $\text{NAND} \times \{0\}$, $\text{NAND} \times \{1\}$ or $\text{NAND} \times \{01\}$ from ϕ . Since ϕ is not mixed Horn, it contains at least one constraint $c = \overline{x_1} \vee \dots \vee \overline{x_k}$ with $k \geq 2$. By Lemma 5 we can obtain the relation \overline{x} or $\overline{x} \wedge y$.

1. By (a) every restraint d other than c contains at least one variable x not from $\{x_1, \dots, x_k\}$; for every such restraint with more than one variable we add the clause \overline{x} resp. $\overline{x} \wedge y$ obtained by Lemma 5, where y is a new variable; this eliminates d by (a). Now all restraints other than c have only one variable.
2. By (b), for every implication $x \vee \overline{y_1} \vee \dots \vee \overline{y_m}$ ($m \geq 1$), at least one y_i or x is not from $\{x_1, \dots, x_k\}$; in the first case identify x with some y_i not from $\{x_1, \dots, x_k\}$, in the second, identify x with any y_i . This eliminates all implications and retains the size of c .
3. If c contains $n > 2$ variables, identify $n - 1$ of them. Further, identify all variables appearing in positive unit clauses, and then all variables appearing in positive unit clauses. Now the formula represents NAND , $\text{NAND} \times \{0\}$, $\text{NAND} \times \{1\}$ or $\text{NAND} \times \{01\}$. \square

Lemma 7. *If \mathcal{S} is a finite set of Horn relations and contains at least one relation that is not mixed Horn, and at least one relation that is not safely componentwise IHSB-, at least one of the relations M , $M \times \{0\}$, $M \times \{1\}$ or $M \times \{01\}$ with $M = (x \vee \overline{y} \vee \overline{z}) \wedge (\overline{x} \vee z)$ is expressible as a $\text{CNF}(\mathcal{S})$ -formula.*

Proof. We modify the proof of Lemma 19 of [15] to work without constants and instead use one of the relations $\{0\}$ or $\{01\}$ obtained by Lemma 5 from the not mixed Horn relation. Let R be some not safely componentwise IHSB- relation from \mathcal{S} .

We skip step 2. Then there may be unit clauses in a CNF-formula representation ϕ of R , but we can obviously eliminate the variables appearing in the unit clauses from the other clauses since they can only take one value in any solution of ϕ . We identify all variables appearing in positive unit clauses, and then all variables appearing in negative unit clauses, then we get a formula ϕ^* that equals ϕ' , $\phi' \wedge x_p$, $\phi' \wedge \overline{x_n}$, or $\phi' \wedge x_p \wedge \overline{x_n}$, with a Horn formula ϕ' without unit clauses and variables x_p and x_n not appearing in ϕ' , depending on whether there were positive, negative, or both unit clauses in ϕ . In the following steps of the proof, we only consider ϕ' and ignore the unit clauses.

In step 6, instead of setting the variables not implied by $\{x, y, z\}$ to 0, we add for every such variable v a clause \overline{v} or $\overline{v} \wedge w$, obtained by Lemma 5 from the not mixed Horn relation of \mathcal{S} . If ϕ^* contained the clause $\overline{x_n}$, we identify v with x_n , if it contained the clause x_p , we identify w with x_p if applicable. This obviously has the same effect on the other clauses of ϕ as setting v to 0, and v does not appear in any non-unit clause any more. In the following steps, we again ignore the unit clauses.

In the last step, we identify the remaining variables other than x, y, z (and those of the unit clauses) with suitable variables from x, y, z instead of setting them to 1: Identify every such variable that has a branch from some variable $b \in \{x, y, z\}$ with b , repeat while there is such a variable. This eliminates all variables other than x, y, z and produces no implications or restraints among x, y, z . \square

3.4 Towards a Trichotomy for $\text{Conn}(\mathcal{S})$: Further Hardness Results

The CNF(\mathcal{S})-formula ϕ' constructed from a CNF_C(\mathcal{S})-formula ϕ in the proof of Lemma 2 using 0- and 1-isolating relations may contain multiple components even if ϕ has only one component. Thus that construction is not appropriate for the connectivity problem; but if we use 0- and 1-unique relations instead, the number of components is retained, and analogously to Lemma 2, we get the following reduction:

Lemma 8. *Let \mathcal{S} be a finite set of logical relations. If there is a 0-unique and an 1-unique CNF(\mathcal{S})-formula, then $\text{CONN}_C(\mathcal{S}) \leq_m^p \text{CONN}(\mathcal{S})$.*

From Lemma 4.13 of [4] we know that if \mathcal{S} contains at least one relation that is not 0-valid, at least one relation that is not 1-valid, and at least one relation that is not complementive², $\overline{x} \wedge y$ is expressible as a CNF(\mathcal{S})-formula, which is both 0- and 1-unique, so we have the following lemma:

² A relation R is complementive if for every vector $(a_1, \dots, a_n) \in R$, also $(a_1 \oplus 1, \dots, a_n \oplus 1) \in R$

Lemma 9. *Let \mathcal{S} be a finite set of logical relations. If \mathcal{S} contains at least one relation that is not 0-valid, at least one relation that is not 1-valid, and at least one relation that is not complementive, $\text{CONN}_{\mathcal{C}}(\mathcal{S}) \leq_m^P \text{CONN}(\mathcal{S})$.*

Thus if \mathcal{S} is not safely tight and satisfies the conditions of the preceding lemma, $\text{CONN}(\mathcal{S})$ is PSPACE-complete. But there are non-safely tight relations that do not satisfy these conditions, and the complexity of $\text{CONN}(\mathcal{S})$ is unknown for sets \mathcal{S} of such relations. It is conceivable that $\text{CONN}(\mathcal{S})$ is in coNP or even in P in such cases; this would mean that there are sets of relations for which $\text{CONN}(\mathcal{S})$ is easier than $\text{ST-CONN}(\mathcal{S})$.

Example 2. The relation $R_{\text{NAE}} = \{0,1\}^3 \setminus \{000,111\}$ is not safely tight but complementive.

Lemma 10. *If \mathcal{S} is a finite set of relations that is safely tight but not Schaefer and contains at least one relation that is not 0-valid, at least one relation that is not 1-valid, and at least one relation that is not complementive, $\text{CONN}(\mathcal{S})$ is coNP-complete.*

Proof. This follows from the correct version (see section 3 in [15]) of Lemma 4.8 of [6] with Lemma 9. \square

Lemma 11. *If \mathcal{S} is a finite set of relations that is not safely tight and contains at least one relation that is not 0-valid, at least one relation that is not 1-valid, and at least one relation that is not complementive, $\text{CONN}(\mathcal{S})$ is PSPACE-complete.*

Proof. This follows from the correct version (see section 3 in [15]) of Theorem 2.8 of [6] with Lemma 9. \square

4 Quantified CSPs

In this section we will examine (partially) quantified $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formulas (in prenex form), i.e. expressions of the form

$$Q_1 y_1 \cdots Q_m y_m \phi(y_1, \dots, y_m, x_1, \dots, x_n),$$

where ϕ is a $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula, and $Q_1, \dots, Q_m \in \{\exists, \forall\}$ are quantifiers. We denote the corresponding connectivity resp. *st*-connectivity problems by $\text{Q-CONN}_{\mathcal{C}}(\mathcal{S})$ resp. $\text{ST-Q-CONN}_{\mathcal{C}}(\mathcal{S})$. As we will see, allowing quantifiers makes the problems harder and the diameter larger in some cases.

4.1 A Dichotomy

Theorem 2. *Let \mathcal{S} be a finite set of logical relations.*

1. *If \mathcal{S} is Schaefer, $\text{ST-Q-CONN}_{\mathcal{C}}(\mathcal{S})$ is in P, $\text{Q-CONN}_{\mathcal{C}}(\mathcal{S})$ is in coNP and the diameter of every quantified $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula is linear in the number of free variables.*

2. Otherwise, both $\text{ST-Q-CONN}_C(\mathcal{S})$ and $\text{Q-CONN}_C(\mathcal{S})$ are PSPACE-complete, and there are quantified $\text{CNF}_C(\mathcal{S})$ -formulas with a diameter exponential in the number of free variables.

Proof. 1. If \mathcal{S} is Schaefer, at least one of the following conditions holds:

- (a) every relation in \mathcal{S} is bijunctive,
- (b) every relation in \mathcal{S} is Horn,
- (c) every relation in \mathcal{S} is dual-Horn,
- (d) every relation in \mathcal{S} is affine.

In the first three cases, any $\text{CNF}_C(\mathcal{S})$ -formula ϕ is itself bijunctive, resp. Horn, resp. dual-Horn. Since these three properties can be characterized by the closure under non-constant operations [5], it follows from Lemma 12 that any relation obtained by arbitrarily quantifying over some variables is still bijunctive, resp. Horn, resp. dual-Horn. Now since by Lemma 4.2 of [6] every bijunctive relation is componentwise bijunctive, and every Horn resp. dual-Horn relation is OR-free resp. NAND-free, the structural properties stated in Lemmas 4.3 and 4.5 of [6] apply in these cases for quantified formulas also. Because the evaluation problem for quantified bijunctive, Horn, and dual-Horn formulas is in P [14], the statements follow from Corollaries 4.4 and 4.6 of [6].

If every relation of \mathcal{S} is affine, we can use the polynomial-time algorithm from Figure 4.2 of [2] to transform any quantified $\text{CNF}_C(\mathcal{S})$ formula ϕ into an equivalent affine formula ϕ' without any quantifiers, and the statements follow from Theorems 2.8, 2.9 and 2.10 of [6].

2. By Schaefer's "expressibility theorem" (Theorem 3.0 of [14]), if \mathcal{S} is not Schaefer, every Boolean relation is expressible from \mathcal{S} by existentially quantifying over some $\text{CNF}_C(\mathcal{S})$ formula, and thus the statements follow from the Lemmas 3.6 and 3.7 of [6]. \square

Lemma 12. *Let R be a logical relation that is closed under the coordinate-wise application of some operation f . Then*

1. *The relation obtained by quantifying existentially over some variable of R is also closed under f .*
2. *If f is not constant, the relation obtained by quantifying universally over some variable of R is also closed under f .*

Proof. 1. Let R be a $n+1$ -ary relation, consisting of m vectors $(a^i, b_1^i, \dots, b_n^i)$, $i = 1, \dots, m$ that is closed under the coordinate-wise application of the k -ary relation f , i.e.

$$(f(a^{i_1}, \dots, a^{i_k}), f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})) \in R$$

for all $1 \leq i_1, \dots, i_k \leq m$. Let the relation $R' = \exists x R(x, \mathbf{y})$ be obtained w.l.o.g by quantifying existentially over the first variable. If then $\mathbf{b}^{i_1}, \dots, \mathbf{b}^{i_k} \in R'$, also $(f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})) \in R'$ since for each $\mathbf{b}^i \in R'$, $(0, \mathbf{b}^i) \in R$ or $(1, \mathbf{b}^i) \in R$, thus R also contains

$$(0, f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k}))$$

or

$$(1, f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})).$$

2. Let R and f be as in (a), but f not constant, and $R' = \forall x R(x, \mathbf{y})$. If then $\mathbf{b}^{i_1}, \dots, \mathbf{b}^{i_k} \in R'$, also $(f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})) \in R'$: Since f is not constant, we can chose values $a_0^{i_1}, \dots, a_0^{i_k} \in \{0, 1\}$ such that $f(a_0^{i_1}, \dots, a_0^{i_k}) = 0$, and $a_1^{i_1}, \dots, a_1^{i_k} \in \{0, 1\}$ such that $f(a_1^{i_1}, \dots, a_1^{i_k}) = 1$. Since for each $\mathbf{b}^i \in R'$ both $(0, \mathbf{b}^i) \in R$ and $(1, \mathbf{b}^i) \in R$, R also contains both

$$(f(a_0^{i_1}, \dots, a_0^{i_k}), f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})) = (0, f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k}))$$

and

$$(f(a_1^{i_1}, \dots, a_1^{i_k}), f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})) = (1, f(b_1^{i_1}, \dots, b_1^{i_k}), \dots, f(b_n^{i_1}, \dots, b_n^{i_k})).$$

□

4.2 Towards a Trichotomy for $\text{Q-CONN}(\mathcal{S})$

$\text{Q-CONN}(\mathcal{S})$ is coNP-complete for Horn not safely componentwise IHSB- sets \mathcal{S} (Lemma 13 of [15]); for $\text{Q-CONN}_C(\mathcal{S})$ we can extend the coNP-complete class, but unfortunately have to reduce the polynomial-time solvable class even more, so that for some sets of relations, the complexity is unknown as of yet.

Definition 5. A logical relation R is Q-safely componentwise IHSB- (Q-safely componentwise IHSB+) if R and every relation R' obtained from R by arbitrary quantification and identification of variables is componentwise IHSB- (componentwise IHSB+).

Theorem 3. Let \mathcal{S} be a finite set of logical relations.

1. If \mathcal{S} is bijective, IHSB-, IHSB+ or affine, $\text{Q-CONN}_C(\mathcal{S})$ is in P.
2. If \mathcal{S} is Horn and not Q-safely componentwise IHSB- or dual Horn and not Q-safely componentwise IHSB+, $\text{Q-CONN}_C(\mathcal{S})$ is coNP-complete.

Proof. 1. This follows from Lemmas 14, 13 and 15 below.

2. This follows from Lemmas 18 and 19 of [15] (in the first step of the proof of Lemma 19, obtaine the not componentwise IHSB- relation by quantification and identification of variables). □

Remark 1. It is not known whether there exists a polynomial-time algorithm if \mathcal{S} is Horn and Q-safely componentwise IHSB- but not IHSB-; it is not possible to reduce $\text{Q-CONN}_C(\mathcal{S})$ to $\text{Q-CONN}_C(\mathcal{S}')$ with a set \mathcal{S}' of IHSB- relations as in the last break of the proof of Lemma 4.13 in [6] for the not quantified case. The obstacle is that the quantifiers in a quantified $\text{CNF}_C(\mathcal{S})$ -formula are applied to the whole formula, not the individual constraints: Suppose $R(x, \mathbf{y}) \in \mathcal{S}$ is Q-safely componentwise IHSB- but not IHSB-, and consider a formula ϕ using R as constraint, $\phi = \phi'(x, \mathbf{y}) \wedge R(x, \mathbf{y})$. Assume there exist solutions $1 \cdot \mathbf{a}$ and $0 \cdot \mathbf{b}$ of ϕ in different components of R , but \mathbf{a} and \mathbf{b} are connected in $G(\exists x R(x, \mathbf{y}))$.

Now consider the formula $\phi_{\exists} = \exists x (\phi'(x, \mathbf{y}) \wedge R(x, \mathbf{y}))$; then \mathbf{a} and \mathbf{b} may or may not be connected in $G(\phi_{\exists})$.

For example, the relation $R = (x \vee \overline{y} \vee \overline{z}) \wedge (\overline{x} \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y} \vee z) = ((\overline{y} \vee \overline{z}) \wedge (\overline{z} \vee \overline{x}) \wedge (\overline{x} \vee \overline{y})) \vee (x \wedge y \wedge z)$ is Horn and Q-safely componentwise IHSB- but not IHSB+. For $\phi = \exists z (R(x, y, z) \wedge (\overline{x} \vee \overline{y}))$, $G(\phi)$ is connected, while for $\phi' = \exists z (R(x, y, z) \wedge z)$, $G(\phi')$ is disconnected.

Lemma 13. *If \mathcal{S} is a IHSB- or IHSB+ set of relations, $\text{Q-CONN}_{\text{C}}(\mathcal{S})$ is in P.*

Proof. We proof the IHSB- case, the IHSB+ case is analogous. We modify the algorithm from the proof of Lemma 4.13 of [6]. First note that any $\text{CNF}_{\text{C}}(S)$ formula is itself IHSB-, and since IHSB- relations can be characterized by the closure under the coordinate-wise application of $f(x, y, z) = x \wedge (y \vee z)$ (see [6] below Definition 4.11), the relation obtained by arbitrarily quantifying over some variables is still IHSB- by Lemma 12. If we could transform the quantified formula ϕ into an equivalent formula ϕ' without quantifiers in polynomial time, we could simply apply the algorithm from [6] to ϕ' . This however seems not possible since quantifier elimination in general can result in an exponential (in the number of quantified variables) increase of the formula size, even for Horn formulas, and this seems also to apply to IHSB- formulas.

However, there exists a polynomial time algorithm to transform any quantified Horn formula into an equivalent Horn formula with only existential quantifiers [3], and we can modify the algorithm from the proof of Lemma 4.13 of [6] (call it A), that decides the connectivity for an unquantified IHSB- formula, to an algorithm A' that works on an existentially quantified one. So we first apply the algorithm of Definition 8 from [3] to ϕ to obtain an equivalent formula ϕ^{\exists} with only existential quantifiers. The matrix of ϕ^{\exists} (the formula obtained by removing the quantifiers) can be written as a conjunction of Horn clauses with all clauses of length greater than 2 containing only negative literals as in [6] since it is still IHSB- as we saw. We also assign all variables of the matrix of ϕ^{\exists} that can take only one value in $\{0, 1\}$ (and remove the corresponding quantifiers in the case on bounded variables) to obtain the formula ϕ'^{\exists} that contains no unit clauses. We can do this easily by checking for each variable whether there exist solutions in which it takes a particular value, since satisfiability of Horn-formulas is decidable in polynomial time.

The modified algorithm A' works on the implication graph $G(\phi'^{\exists})$ of the matrix of ϕ'^{\exists} , defined exactly as in [6]. Let the sets S_j and T_i also be defined as in [6]. The algorithm rejects iff there exists a free variable x_i such that $x_i \in T_i$, T_i does not contain any S_j and furthermore, some directed cycle in which x_i lies contains at least one additional free variable.

To prove the correctness of A' , we show

- (*) A' rejects (executed on the matrix of ϕ'^{\exists}) iff A would reject if it were executed on the formula ϕ' obtained by eliminating all quantifiers from ϕ^{\exists} .

It is clear that ϕ' also will contain no unit-clauses since existential quantification can be seen as the projection along the variables over which is quantified. The

proof is by induction on the number of existential quantifiers, by showing (a) - (d) below.

Therefor first note that since an (unquantified) IHSB- formula ψ is completely determined by its implication graph $G(\psi)$ together with the sets S_j , we just have to consider how the implication graph and the sets S_j change when we quantify existentially over some variable y appearing in $\psi(\mathbf{x}, y)$ (o.b.d.a. we chose y to be the last variable of ψ). We will simply speak of the edge $\bar{x}_i \vee x_j$ to mean the corresponding edge from x_i to x_j , and of the set $\bar{x}_{i_1} \vee \bar{x}_{i_2} \vee \dots$ to mean the corresponding set S_j . Let

$$\psi(\mathbf{x}, y) = c_1(\mathbf{x}, y) \wedge c_2(\mathbf{x}, y) \wedge \dots \wedge d_1(\mathbf{x}) \wedge d_2(\mathbf{x}) \wedge \dots,$$

where c_i resp. d_i are the clauses containing y resp. not containing y . Then eliminating the quantifiers from $\psi'(\mathbf{x}) = \exists y \psi(\mathbf{x}, y)$ gives

$$\psi'(\mathbf{x}) = \psi(\mathbf{x}, 0) \vee \psi(\mathbf{x}, 1) = \left(\bigwedge_{i,j} c_i(\mathbf{x}, 0) \vee c_j(\mathbf{x}, 1) \right) \wedge d_1(\mathbf{x}) \wedge d_2(\mathbf{x}) \wedge \dots$$

So the directed edges between free variables $\bar{x}_i \vee x_j$ are clearly not affected. We consider all possible combinations of IHSB- clauses $c_i(\mathbf{x}, y)$ and $c_j(\mathbf{x}, y)$ containing y (that are of the form $\bar{x}_i \vee y$, $x_i \vee \bar{y}$, or $\bar{y} \vee \bar{x}_{i_1} \vee \bar{x}_{i_2} \vee \dots$) to disjunctions $c_i(\mathbf{x}, 0) \vee c_j(\mathbf{x}, 1)$; combinations where y appears positive resp. negative in both clauses are tautological and can be discarded. For the implication graph, we observe the following:

For the incoming resp. outgoing edges of y , $\bar{x}_i \vee y$ resp. $\bar{y} \vee x_j$, each pair of an outgoing and an incoming edge is replaced a by an edge $\bar{x}_i \vee x_j$; for $i = j$ this clause is tautological and can be removed. The following properties are easy to verify:

- (1) For every free variable x_i , the set of other free variables reachable from it is retained.
- (2) x_i is still reachable from itself iff it lay in a cycle containing at least one additional free variable.
- (3) The sets S_j only containing free variables are not affected,
- (4) For the ones containing y , of the form $\bar{y} \vee \bar{x}_{i_1} \vee \bar{x}_{i_2} \vee \dots$, for each in y incoming edge $\bar{x}_k \vee y$, a set $\bar{x}_k \vee \bar{x}_{i_1} \vee \bar{x}_{i_2} \vee \dots$ is created.
- (5) No other sets S are created.

To show (*), it suffices to show the following

- (a) No new cycles emerge: This is clear by property (1) above.
- (b) Any free variable x_i that lay in some cycle in $G(\psi)$ still lies in a cycle in $G(\psi')$ iff it lay in some cycle in $G(\psi)$ with at least one additional free variable: this is (2).
- (c) For each free variable x_i , if the set T_i of the variables reachable from x_i in $G(\psi)$ contained a set S_j , the set T'_i of the variables reachable from x_i in $G(\psi')$ still contains some set S'_j : We only have to consider the case that S_j

contained y because of (1) and (3); then, a set containing the variable from which y was reachable from x_i (and no variable not reachable from x_i) is created by (4).

- (d) If T_i did not contain any S_j , T'_i still contains no S'_j : Because of (5), this means the following: Any set S_j not contained in T_i does not produce a set contained in T'_i . Now by (1) and (3), we only have to consider the case that S_j contained y as the only variable not reachable from x_i ; then, y either had no incoming edge in which case no set is produced from S_j , or it had incoming edges only from variables not reachable from x_i , in which case all sets produced from S_j contain a variable not reachable from x_i by (4). \square

Lemma 14. *If \mathcal{S} is a bijunctive set of relations, $\text{Q-CONN}_C(\mathcal{S})$ is in P.*

Proof. Makino, Tamaki and Yamamoto show in [11] below Proposition 2 that any bijunctive formula can be transformed in an connectivity-equivalent Horn 2-CNF formula by renaming variables: We can calculate a solution \mathbf{a} in linear time [1] (w.l.o.g. we may assume that a solution exists) and then take $\psi(\mathbf{x}) = \phi(x_1 \oplus a_1, x_2 \oplus a_2, \dots)$. Now ψ is clearly Horn since $\psi(0, \dots, 0) = 1$, and the connectivity is retained since $|(x_1 \oplus a_1, x_2 \oplus a_2, \dots) - (y_1 \oplus a_1, y_2 \oplus a_2, \dots)| = |(x_1, x_2, \dots) - (y_1, y_2, \dots)|^3$.

Since any $\text{CNF}_C(\mathcal{S})$ formula ϕ is itself bijunctive, given $Q_1 y_1 \dots Q_m y_m \phi(\mathbf{x}, \mathbf{y})$, we can instead take $Q_1 y_1 \dots Q_m y_m \psi(\mathbf{x}, \mathbf{y})$, where ψ is the Horn 2-CNF formula obtained from ϕ as described, and then apply the algorithm from the IHSB- case since any Horn 2-CNF formula is also IHSB- (and renaming of quantifier-bounded variables does not make a difference). \square

Lemma 15. *If \mathcal{S} is an affine set of relations, $\text{Q-CONN}_C(\mathcal{S})$ is in P.*

Proof. We can use the polynomial-time algorithm from Figure 4.2 in [2] to transform any quantified $\text{CNF}_C(\mathcal{S})$ formula into an equivalent one without any quantifiers, and then apply the algorithm for $\text{CONN}_C(\mathcal{S})$ from Lemma 4.10 of [6]. \square

5 Open Problems

The complexity of $\text{CONN}(\mathcal{S})$ is open for non-Schaefer sets \mathcal{S} containing only 0-valid, only 1-valid, or only complementive relations. Besides the sets \mathcal{S} considered in Lemma 9, it may be possible to find further ones from which both 0-unique and an 1-unique relations can be expressed as $\text{CNF}(\mathcal{S})$ -formulas, which would by Lemma 8 lead to further hardness results.

As stated in Section 3.4, in the other direction it would be especially interesting if one could find a P- or coNP-algorithm for $\text{CONN}(\mathcal{S})$ for some not safely tight set \mathcal{S} , since this would show that there are cases for which the connectivity problem is easier than the st -connectivity problem.

The complexity of $\text{Q-CONN}_C(\mathcal{S})$ is open for Horn and Q-safely componentwise IHSB- sets that are not IHSB-, and for dual Horn and Q-safely componentwise IHSB+ sets that are not IHSB+.

³ For two Boolean vectors \mathbf{a} and \mathbf{b} , $|\mathbf{a} - \mathbf{b}|$ denotes the Hamming distance, i.e., the number of positions in which they differ

```

[110] -> already 1-isolating
[000 110] -> already 1-isolating
[100 110] identify x1,x2 -> [10]
[000 100 110] take R(x1,x2,x3) AND R(x2,x1,x3) -> [000 110]
[110 001] -> already 1-isolating
[000 110 001] -> already 1-isolating
[100 110 001] -> already 1-isolating
[000 100 110 001] take R(x1,x2,x3) AND R(x2,x1,x3) -> [000 110 001]
[110 101] -> already 1-isolating
[000 110 101] -> already 1-isolating
[100 110 101] identify x1,x2 -> [10]
[000 100 110 101] take R(x1,x2,x3) AND R(x2,x1,x3) -> [000 110]
[110 001 101] -> already 1-isolating
[000 110 001 101] -> already 1-isolating
[100 110 001 101] identify x1,x2 -> [10 01]
[000 100 110 001 101] take R(x1,x2,x3) AND R(x2,x1,x3) -> [000 110 001]
[110 011] -> already 1-isolating
[000 110 011] -> already 1-isolating
[100 110 011] -> already 1-isolating
[000 100 110 011] -> already 1-isolating
[110 001 011] -> already 1-isolating
[000 110 001 011] -> already 1-isolating
[100 110 001 011] identify x1,x2 -> [10 01]
[000 100 110 001 011] take R(x1,x2,x3) AND R(x1,x3,x2) -> [000 100 011]
[110 101 011] -> already 1-isolating
[000 110 101 011] -> already 1-isolating
[100 110 101 011] -> already 1-isolating
[000 100 110 101 011] -> already 1-isolating
[110 001 101 011] -> already 1-isolating
[000 110 001 101 011] -> already 1-isolating
[100 110 001 101 011] identify x1,x2 -> [10 01]
[000 100 110 001 101 011] take R(x1,x2,x3) AND R(x1,x3,x2) -> [000 100 110 101 011]
[110 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 110 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[100 110 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 100 110 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[110 001 111] -> already 1-isolating
[000 110 001 111] identify x1,x3 -> [00 11]
[100 110 001 111] -> already 1-isolating
[000 100 110 001 111] identify x1,x3 -> [00 11]
[110 101 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 110 101 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[100 110 101 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 100 110 101 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[110 001 101 111] identify x1,x3 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 110 001 101 111] identify x1,x3 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[100 110 001 101 111] identify x1,x3 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 100 110 001 101 111] identify x1,x3 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[110 011 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 110 011 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[100 110 011 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 100 110 011 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[110 001 011 111] identify x1,x3 -> [11]
[000 110 001 011 111] identify x1,x3 -> [00 11]
[100 110 001 011 111] identify x1,x3 -> [11]
[000 100 110 001 011 111] identify x1,x3 -> [00 11]
[110 101 011 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 110 101 011 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[100 110 101 011 111] identify x1,x2 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 100 110 101 011 111] identify x1,x2 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[110 001 101 011 111] identify x1,x3 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 110 001 101 011 111] identify x1,x3 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]
[100 110 001 101 011 111] identify x1,x3 -> [10 11] take R(x1,x2) AND R(x2,x1) -> [11]
[000 100 110 001 101 011 111] identify x1,x3 -> [00 10 11] take R(x1,x2) AND R(x2,x1) -> [00 11]

```

Fig. 1. Producing a 1-isolating relation from every 3-ary relation R satisfying $110 \in R$ and $010 \notin R$ for the last case of the proof of Lemma 1.

References

1. Aspvall, B., Plass, M.F., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.* 8(3), 121–123 (1979), [http://dx.doi.org/10.1016/0020-0190\(79\)90002-4](http://dx.doi.org/10.1016/0020-0190(79)90002-4)
2. Bauland, M.: Complexity results for boolean constraint satisfaction problems. Ph.D. thesis (2007)
3. Bubeck, U., Büning, H.K.: Models and quantifier elimination for quantified horn formulas. *Discrete Applied Mathematics* 156(10), 1606–1622 (2008), <http://dx.doi.org/10.1016/j.dam.2007.10.005>
4. Creignou, N., Hermann, M.: Complexity of generalized satisfiability counting problems. *Information and Computation* 125(1), 1–12 (1996)
5. Creignou, N., Khanna, S., Sudan, M.: Complexity classifications of boolean constraint satisfaction problems. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2001)
6. Gopalan, P., Kolaitis, P.G., Maneva, E., Papadimitriou, C.H.: The connectivity of boolean satisfiability: Computational and structural dichotomies. *SIAM J. Comput.* 38(6), 2330–2355 (Mar 2009), <http://dx.doi.org/10.1137/07070440X>
7. Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H.: The connectivity of boolean satisfiability: Computational and structural dichotomies. pp. 346–357. *ICALP’06* (2006), http://dx.doi.org/10.1007/11786986_31
8. Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H.: The connectivity of boolean satisfiability: Computational and structural dichotomies. *CoRR abs/cs/0609072* (2006), <http://arxiv.org/abs/cs/0609072>
9. Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. *Theor. Comput. Sci.* 412(12–14), 1054–1065 (Mar 2011), <http://dx.doi.org/10.1016/j.tcs.2010.12.005>
10. Kamiński, M., Medvedev, P., Milanič, M.: Shortest paths between shortest paths and independent sets. In: *Combinatorial Algorithms*, pp. 56–67. Springer (2011)
11. Makino, K., Tamaki, S., Yamamoto, M.: On the boolean connectivity problem for horn relations. In: *Proceedings of the 10th international conference on Theory and applications of satisfiability testing*. pp. 187–200. *SAT’07* (2007)
12. Maneva, E., Mossel, E., Wainwright, M.J.: A new look at survey propagation and its generalizations. *Journal of the ACM (JACM)* 54(4), 17 (2007)
13. Mézard, M., Mora, T., Zecchina, R.: Clustering of solutions in the random satisfiability problem. *Physical Review Letters* 94(19), 197205 (2005)
14. Schaefer, T.J.: The complexity of satisfiability problems. pp. 216–226. *STOC ’78* (1978)
15. Schwerdtfeger, K.W.: A computational trichotomy for connectivity of boolean satisfiability. *ArXiv CoRR abs/1312.4524* (2013), <http://arxiv.org/abs/1312.4524>, extended version of a paper submitted to the *JSAT Journal*